Factor Analysis and Multivariate Regression

Humboldt-Universität zu Berlin
School of Business and Economics
Institute for Statistics and Econometrics
Ladislaus von Bortkiewicz Chair of Statistics

by

Daria Fitisova (578593)
Maria Kozlova (561736)
Yihan Liu (577959)
Andrea Mina Weihe (554602)

Berlin, August 4, 2016

Contents

List of Figures

List of Tables

1 Introduction

There are 24 hours in a day, and everyone is free to distribute this number on different occupations. The goal of current project is to determine, how leisure, work and time spent on personal needs correspond to each other. Namely, we set ourselves the following goals:

1. Scrutinize time distribution in general: estimate the absolute values and proportions (data visualization task).

2. Study the relations among activities (will be solved by multivariate regression application.

3. Extract groups of activities out of given ones (factor analysis issue)

The aims stated above are solved while analyzing the "Timebudget"dataset, which was firstly used by M.Volle in the book "Analyse des Donnes"(1985). The data was collected in 1976 and represents the time distribution of 28 person during 100 days. There are 10 kinds of occupation given: professional activity (PROF), transportation linked to professional activity (TRAN), household occupation (HOUS), occupation linked to children (KIDS), shopping (SHOP), time spent on personal care (PERS), eating (EAT), sleeping (SLEE), watching television (TELE), other leisures (LEAS). As it was stated above, in general the current paper includes three sections: data visualization, multivariate regression application and factor analysis execution. It is needed to mention, that the analysis is entirely done in R-package. The following codes as well as the corresponding interpretation and conclusions are stated in the paper further.

1. Visualization

2. Regression

3. Factor analysis

The next goal we set ourselves is to reduce the dimension of the observed data, detecting a structure in the relationship between variables. Namely, we intend to reproduce the information, contained in initial "timebudget" dataset by a smaller number of factors. This aim corresponds to the question: are there any latent characteristics by which the given kinds of occupation can be grouped? The target mentioned above can be solved by the factor analysis (FA) implementation, which will be represented in the following part of our project.

2 Exploratory Data Analysis

We start our data analysis task with taking a look at the data frame at hand. Exploratory data analysis is an approach that usually consists of summarizing the main characteristics of the data frame, often with the help of visual techniques, in order to give the researcher a brief overview of existing variables and help determine the direction of further research. Visualization of data is widely applied in the context of preliminary data analysis. The goal of graphs and plots is to communicate information in a clear and efficient manner, helping the users to save time and efforts of manually searching through the data. We first start with the three simplest statistical commands providing basic information regarding the dataset, namely '*summary( )*', '*boxplot( )*'and '*histogram( )*'. The *summary( )* command provides us with the six most important descriptive statistics of variables'distributions: minimum, maximum, mean, median and two quantiles first and second necessary for calculation of the interquartile range.

```
> summary(Timebudget)
```

PROF	TRAN	HOUS	KIDS	SHOP
Min.:10.0	Min.:10.0	Min.:50.0	Min.:10.0	Min.:52.0
1st Qu.: 386.8	1st Qu.: 47.50	1st.: 96.5	1st Qu.: 10.00	1st Qu.: 85.0
Median: 535.5	Median: 95.50	Median: 256.0	Median: 22.00	Median: 112.0
Mean: 450.5	Mean: 86.11	Mean: 277.1	Mean: 32.32	Mean: 108.8
3rd Qu.: 631.0	3rd Qu.: 127.00	3rd Qu.: 424.0	3rd Qu.: 56.00	3rd Qu.: 131.8
Max.: 656.0	Max.: 148.00	Max.: 710.0	Max.: 110.00	Max.: 170.0

PERS	EAT	SLEE	TELE	LEAS
Min.:77.0	Min.:85.0	Min.:745.0	Min.:40.0	Min.:228.0
1st Qu.: 90.00	1st Qu.: 100.0	1st.: 762.2	1st Qu.: 64.75	1st Qu.: 308.8
Median: 92.00	Median: 111.0	Median: 775.0	Median: 91.50	Median: 347.0
Mean: 94.93	Mean: 118.1	Mean: 785.9	Mean: 99.43	Mean: 345.8
3rd Qu.: 96.25	3rd Qu.: 132.5	3rd Qu.: 809.2	3rd Qu.: 122.75	3rd Qu.: 385.8
Max.: 130.00	Max.: 180.0	Max.: 849.0	Max.: 180.00	Max.: 475.0

The first look at the numbers already gives us some hints about the data: the ranges of variables vary dramatically, with PERS (personal time) having an IQR of only 6.25 and

PROF (professional activity) of 244.2 hours. We use the '*boxplot( )*'command to visualize the findings in a more readily understandable format:

```
> boxplot(timebudge, col = topo.colors(10), ylab="time")
```

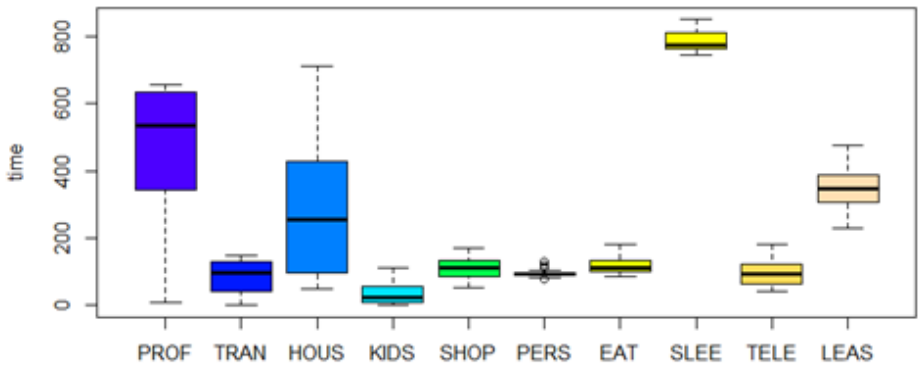

Figure 1: Boxplot of Timebudget

The plot shows again, that ranges and medians differ dramatically over the set of variables, with most time spent on average on sleep (Median: 775.0, Mean: 785.9), and much less time spent on additional life activities, such as shopping, eating and caring for children. The PERS variable seems to have several outliers points which lie outside of the 1.5*IQR demarcation line set by John Tukey. We can also notice that some variables exhibit a skewed distribution, which is especially strong with PROF and HOUS.

We proceed further with our visual analysis by constructing histograms of our variables with the help of the '*ggplot2*'package. The '*ggplot()*'command is a powerful tool for creating elegant and complex graphs in R. The '*ggplot2* functions can be elaborated with '+'signs and the assigned 'aes'*(aesthetic)* which generates a mapping of the variables to certain parts of the graph.

The code is written as follows: we start by opening a pdf-device in R and creating a named file that will be 'filled'by the following 'ggplot'command. The '*geom_histogram()*'adds a histogram to the mapped area, and '*geom_density()*'adds a kernel density estimate useful for displaying the distribution of depicted variables with underlying smoothness. The operation is finished by 'dev.off()'that closes and saves the file in the specified directory.

```
pdf(file = "histPROF.pdf")
par(mfrow=c(1,1))
ggplot(data=Timebudget, aes(Timebudget$PROF)) +
        geom_histogram(aes(y =..density..),
                       breaks=seq(0, 700, by = 20),
                       col="black",
                       fill="blue",
                       alpha = .2) +
        geom_density(col=2) +
        labs(title="Professional Activity") +
        labs(x="Hours", y="Count")
dev.off()
```

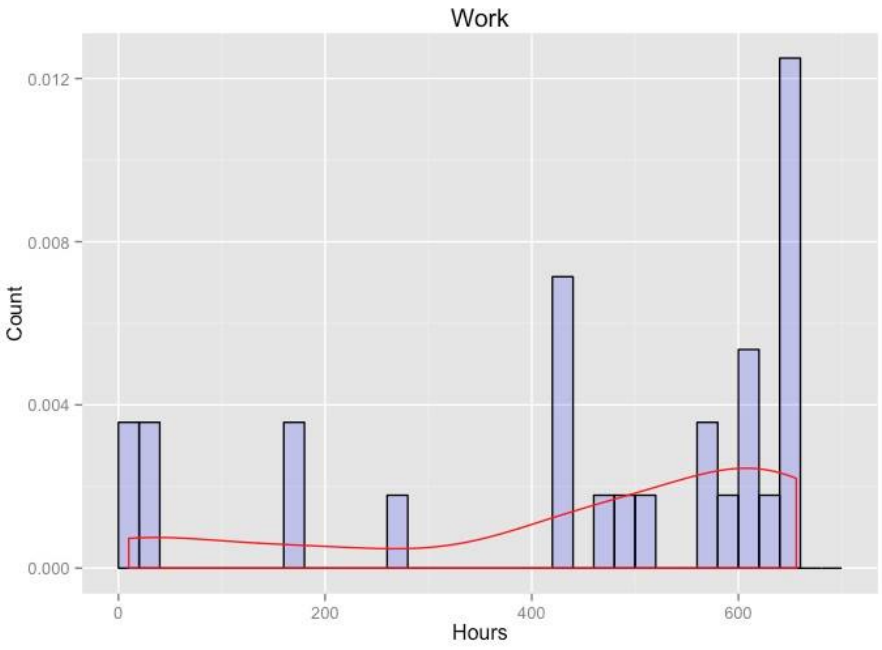

Figure 2: test

For our convenience, and for visual clarity, we set the bin sizes and the x-axis lengths in the graphs being equal for all histograms, and drop the density lines for the time being. This would allow an easier comparison of the variables.

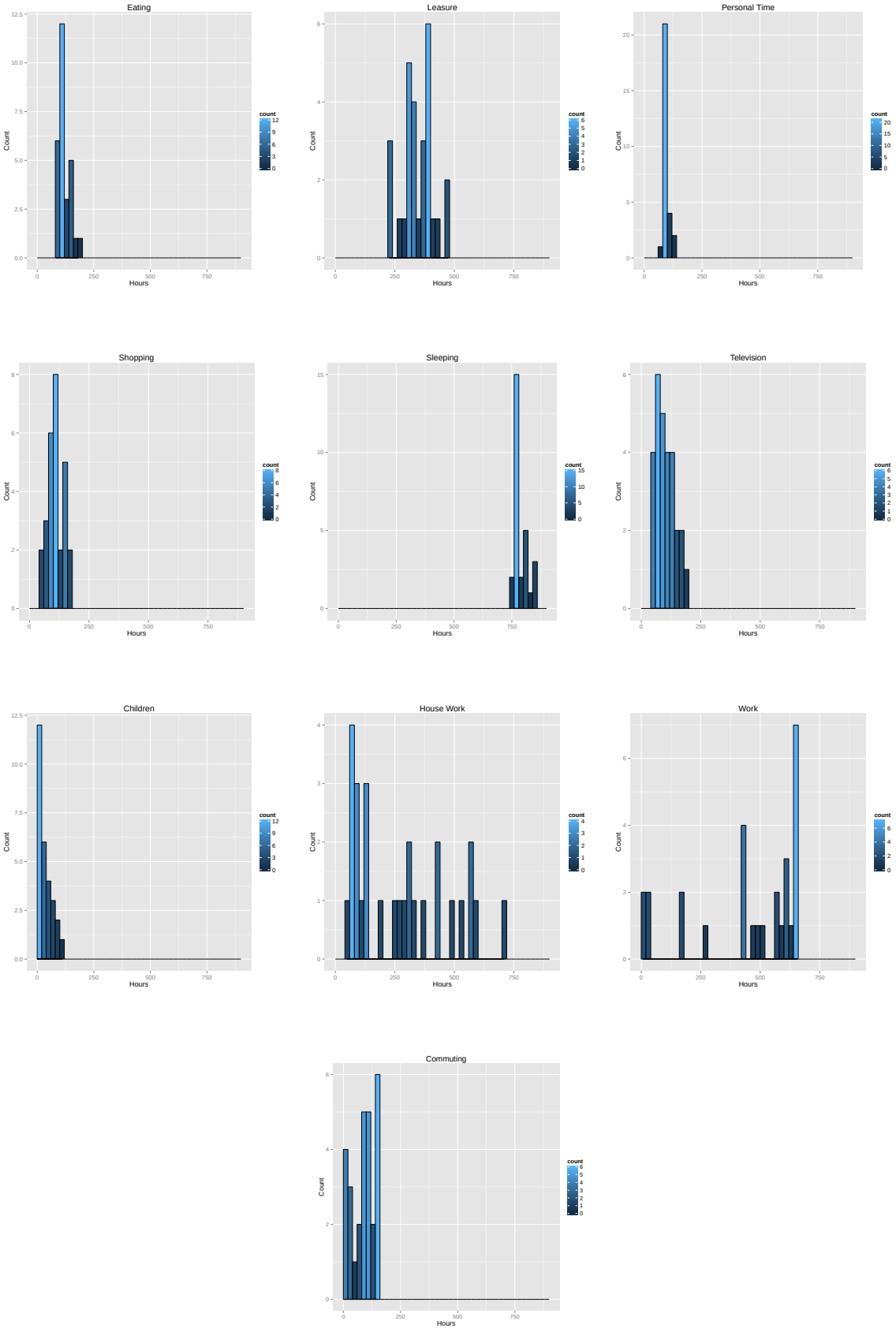

Figure 3: Histograms

We can see that most variables (EAT, LEAS, PERS, SHOP, SLEE, TELE) have an approximately normal distribution. Other variables, such as KIDS, HOUS, PROF, are somewhat skewed. It is important to keep in mind that the sample size of our data is extremely small (28 observations), so it is not unusual to observe skewed distributions in some variables due to chance alone.

The histograms present a very clear picture of the distribution of hours spent on each activity, but to get a more precise look we employ kernel density estimates as separate plots.

```
par(mfrow=c(3,3),mar=c(2,1,1,1))
dfplot <- function(data.frame) {
        df <- data.frame
        ln <- length(names(data.frame))
        for(i in 1:ln){
                plot(density(df[,i],main=names(df)[i]),
                main= colnames(df)[i])
        }
}
dfplot(Timebudget)
```

par(mfrow=c(3,3),mar=c(2,1,1,1)) dfplot ¡- function(data.frame) df ¡- data.frame ln ¡- length(names(data.frame)) for(i in 1:ln) plot(density(df[,i],main=names(df)[i]), main= colnames(df)[i]) dfplot(Timebudget)

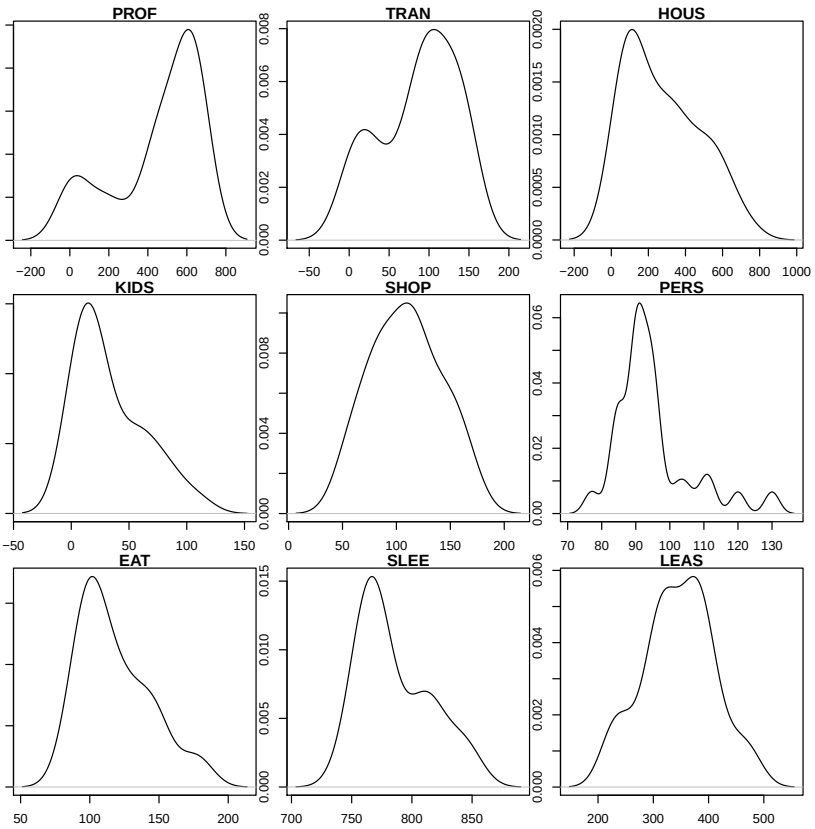

Figure 4: test

The density plots are more smooth than histograms, and show us a more generalized picture of the variables‘ distributions. Taking into account the previously mentioned obstacle of a small sample size, we can conclude that most variables (with the exception of PERS, PROF and TRAN) are close to a normal distribution. We may take a risk and assume that skewed distributions would vanish with a large enough sample size. Keeping these findings in mind, we move on to apply multivariate regression analysis on our data sample in Section II.

3 Factor Analysis

3.1 Principal Component Analysis

The very first step in performing factorization methods is to understand, can the dimensionality be in general reduced? If yes, how many components are reasonable and sufficient to use? We perform principal component analysis (PCA) to answer the stated questions. PCA can be easily done in R via princomp() function:

```
pca.time=princomp(time, scores=T, cor=T)
summary(pca.time)
```

By means of summary() command we get the following information about the importance of components:

	Comp.1	**Comp.2**	**Comp.3**	**Comp.4**	**Comp.5**	**...**	**Comp.10**
Standard Deviation	2.14	1.46	0.15	1.09	0.68	...	0
Proportion of variance	0.46	0.21	0.13	1.12	0.05	...	0
Cumulative proportion	0.46	0.67	0.80	0.92	0.97	...	1

Table 1: Importance of the Components

According to table 2, the first component explains 46% of the initial data variance. However, cumulative proportion of the fluctuations, which can be described by the first two components, is 0.67. In the same manner, the more components we take into account, the more variance of the primary data can be explained. The question is: what is the optimal number of factors to be considered? There is an informal rule: the components, whose contribution into amount of explained variance is not notable in comparison with others, should not be considered. This rule can be visualized by the scree plot:

```
plot(pca.time, type="l")
```

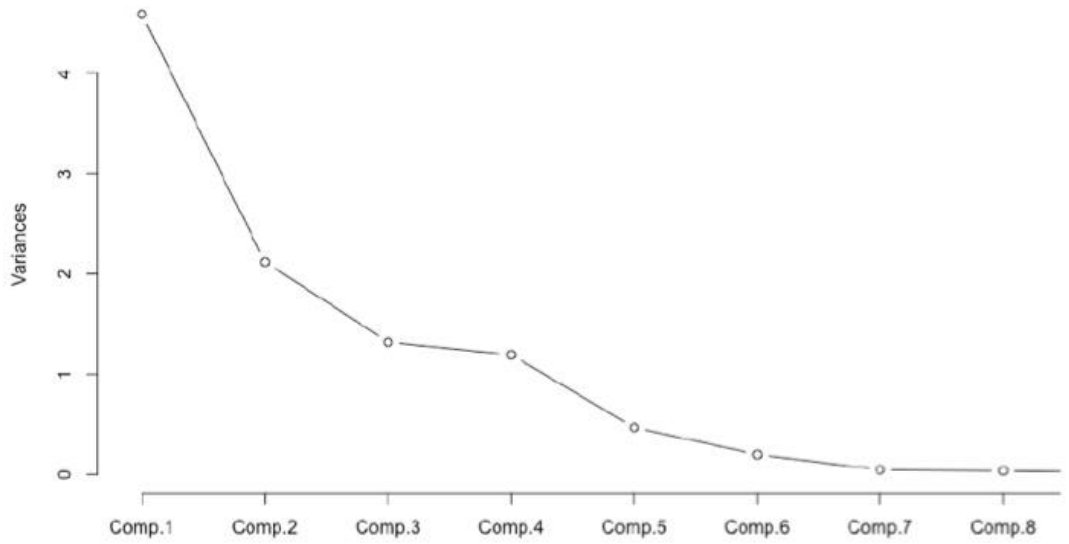

Figure 5: PCA Screeplot

It follows from Figure 4, that proportion of the explained variance doesn't change crucially since the fifth component. In this way it would be logical to assume four-components model. Nevertheless, it is also clear, that there is practically no difference between third and fourth factors eigenvalues. As far as our aim is to reduce the dimension, we find reasonable to consider a model with 3 components. As soon as the number of components is defined, it is useful to pay attention to the interpretational aspect. In other words, we should check, whether the components differ from each other (is every component unique and requisite?). The easiest way to evaluate the descriptive consistency of each component is to visualize the vectors of initial variables in the principal components space. It can be done with the help of biplot() function, where the "choices" subcommand is used to define dimensions:

```
bip1=biplot(pca.time, choices=1:2)
```

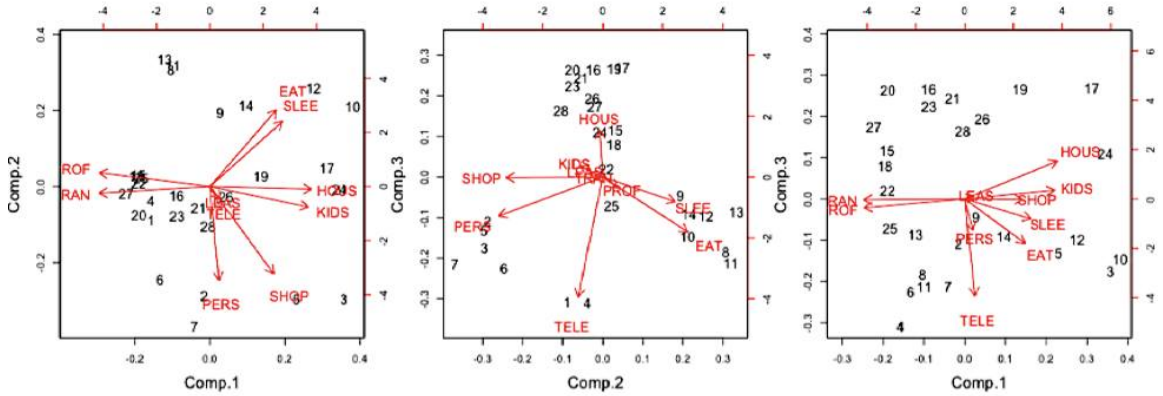

Figure 6: Vectors of initial variables in principal components space

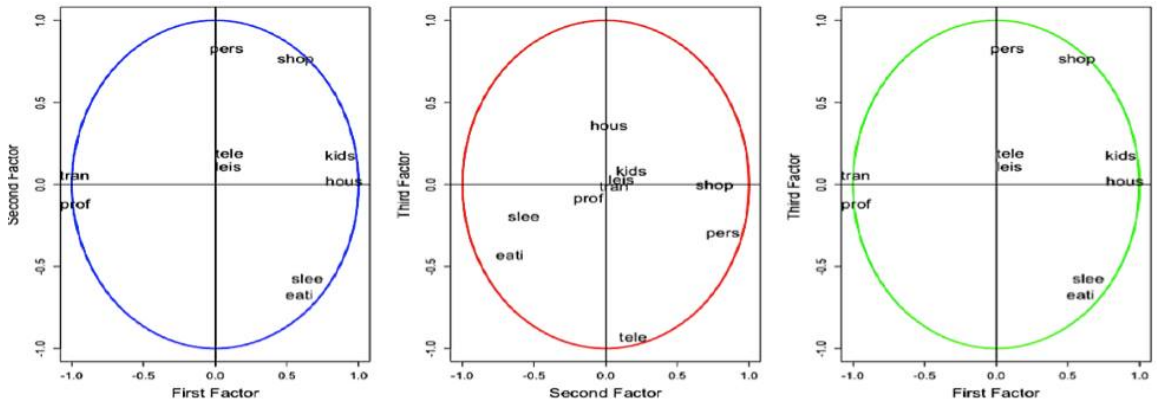

Figure 7: Representation of the variables in principal components space[1]

As it is shown on the figures 5 and 6, the components are really unique, comprising different initial variables. At this point we will not go deep into interpretation, because our primary goal is to run FA, using PCA just as a tool for factors number estimation. Nevertheless, it is informatively useful to take a look at the observations distribution within the principal components space. For this purpose we have used "rgl"and "pca3d"packages. They allow to generate interactive plot, which is a crucial benefit for 3-d space representation.

[0]We did not paste the corresponding code intentionally (because of its complexity and interchangeability with simple biplot() function). When necessary, it can be found in the enclosed R-file.

```
library(rgl)
plot3d(pca.time$scores[,1:3])
library(pca3d)
pca3d(pca.time, components = 1:3)
```

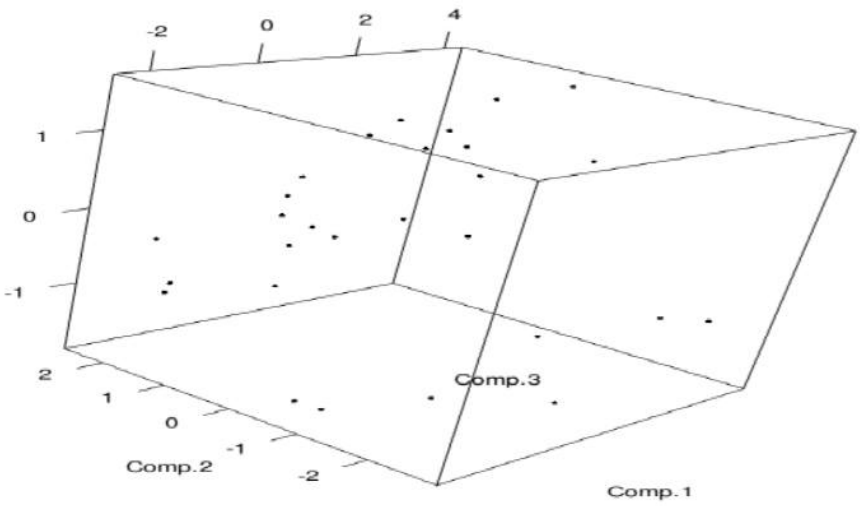

Figure 8: Representation of the observations in principal components space *(via plot3d())*

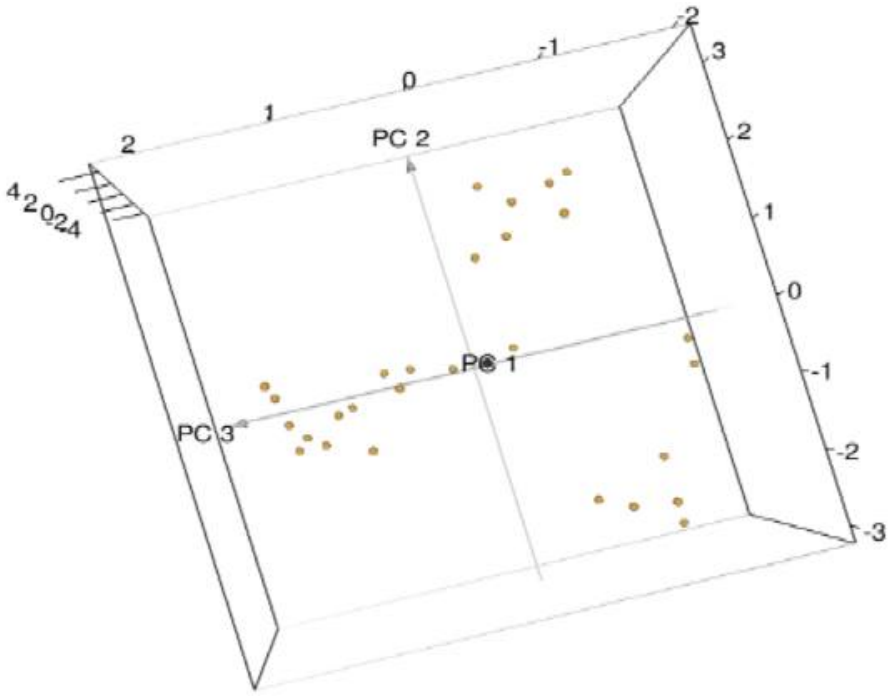

Figure 9: Representation of the observations in principal components space *(via pca3d())*

The visualization of observations distribution within the principal components space can be treated as a first step to cluster analysis implementation. As it follows from Figure 8, the data can be possibly grouped into three clusters.

3.2 Factor Analysis

Basing on a result of PCA in this section we will introduce the results of factor analysis. The least can be accomplished with the help of factanal() function.

```
fa.time=factanal(time2, factors=3, rotation="none")
```

We should mention, that in order to implement FA we had to transform initial dataset. The reason is that function factanal() internally uses solve() command which is a numerical way to calculate the matrix inverse. As far as some of the numbers in "timebudget" dataset are very small (whereas the others are too big), it assumes that they least are zero, leading to the assumption that the matrix is singular. To solve this problem we have taken the squared root from all of the observations: *time2=sqrt(time)* On the very first step we use "simple" FA, with no rotation implementation. As the result the factor loadings are shown (which can be also called by command *load=fa.time$loadings)*:

	Factor 1	**Factor 2**	**Factor 3**
Professional activity	-0.85	0.26	-0.39
Transportation linked to professional activity	-0.92	0.14	-0.35
Household occupation	0.85	-0.50	
Occupation linked to children	0.79	-0.40	
Shopping	0.27	-0.60	0.53
Time spent on personal care	-0.14	-0.38	0.40
Eating	0.76	0.64	
Sleeping	0.73	0.41	
Watching Television		0.49	0.37
Other leisures			0.73

Table 2: Factor loadings before rotation

It is clear from the table 2 that factors we got are not so much distinct. In other words, there are many variables, which have big factor loadings in several factors. This seriously hardens the interpretation process. In order to get more contrast factors we may use the rotation option. In order to decide, which rotation function to use, we should check, whether factors are correlated or not. It can be done by plotting the distribution of loadings pairwise.

```
pairs(load)
```

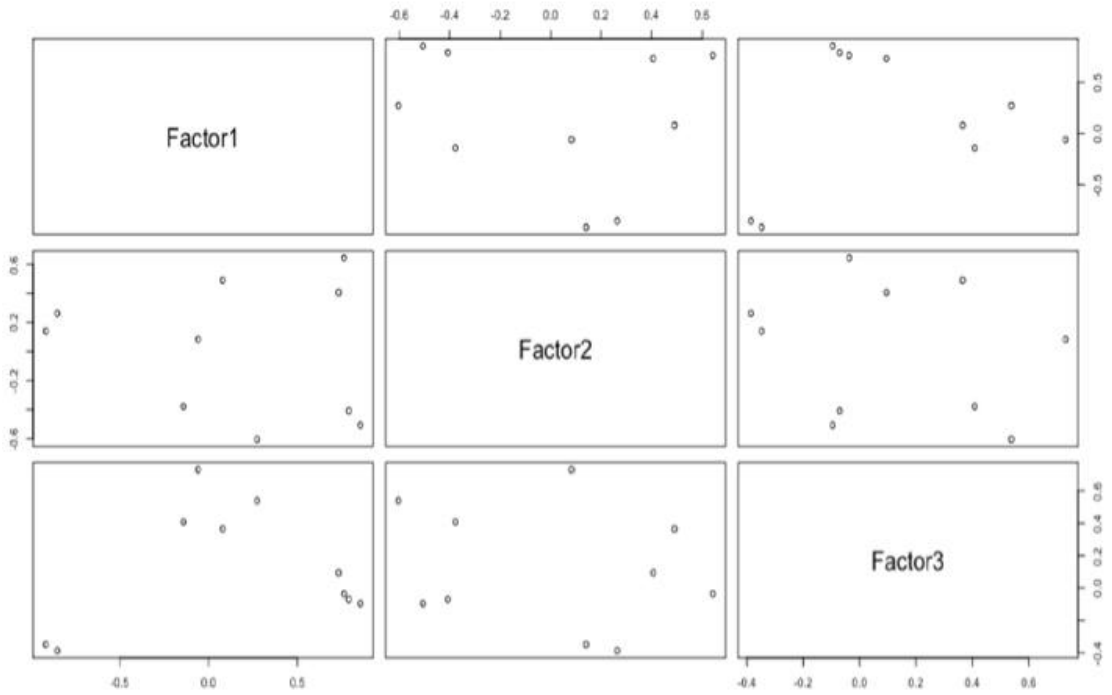

Figure 10: Distributions of factor loadings

It comes from the figure above, that there is no strong correlation between loadings distributions. Based on this fact we can conclude, that appropriate in this way rotation is orthogonal. We have used "varimax".

```
fa.time.rot=factanal(time2, factors=3, rotation="varimax")
load.rot=fa.time.rot$loadings
```

A nice feature of factanal() function is a test of factors number sufficiency. In our case p-value for chi-squared test is 2.57e-11 (>0.05), which means that the null hypothesis H0: "three factors are not sufficient" should be rejected.

	Factor 1	**Factor 2**	**Factor 3**
Professional activity	-0.90	-0.27	-0.26
Transportation linked to professional activity	-0.90	-0.37	-0.16
Household occupation	0.98	-0.18	
Occupation linked to children	0.88	-0.11	
Shopping	0.50	-0.10	0.69
Time spent on personal care		-0.13	0.56
Eating	0.45	0.73	-0.51
Sleeping	0.52	0.60	-0.28
Watching Television	-0.11	0.61	
Other leisures		0.43	0.59

Table 3: Factor loadings after varimax rotation

	Factor 1	**Factor 2**	**Factor 3**
SS loadings	4.10	1.73	1.574
Proportion variance	0.41	0.17	0.16
Cumulative variance	0.41	0.58	0.74

Table 4: Proportion of variance explained after varimax rotation

According to the table 4, three factors explain 74% of the initial data variation. We treat it as a good result. Another good point to consider is that factor loadings became more "contrast" after the use of varimax rotation. The result is still not excellent, but to some extent easier to interpret. To make the task of interpretation not so complex, we can represent initial variables in factors space. It can be done via *pairs(load)* function:

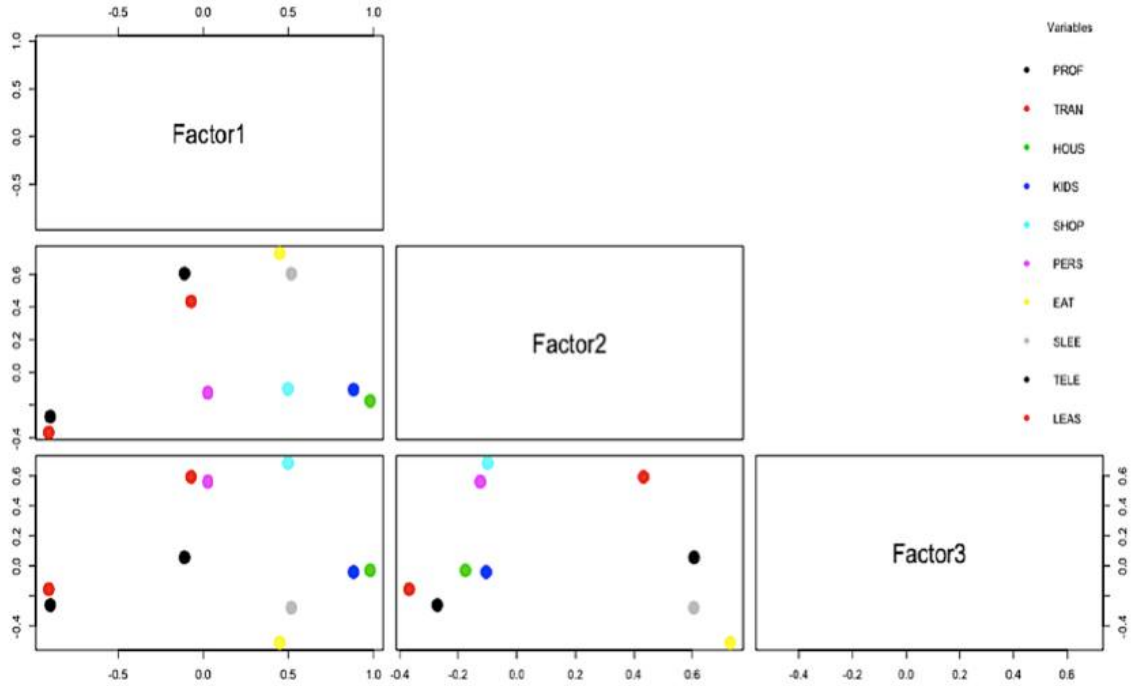

Figure 11: Representation of the initial variables in factors space (via *pairs()* function)

or, for more detailed plot, using a code (example for factor 1 factor 2 space)[1]

```
load=fa.time$loadings[,1:2]
plot(load,type="n",xlab="Factor 1",ylab="Factor 2")
text(load,labels=names(time2),cex=.7)
abline(h=0,v=0)
```

[1]It should be mentioned, that for the space of 1st and 3rd factors the first line of the code should be: *load.rot3=cbind(fa.time.rot$loadings[,1],fa.time.rot$loadings[,3])*

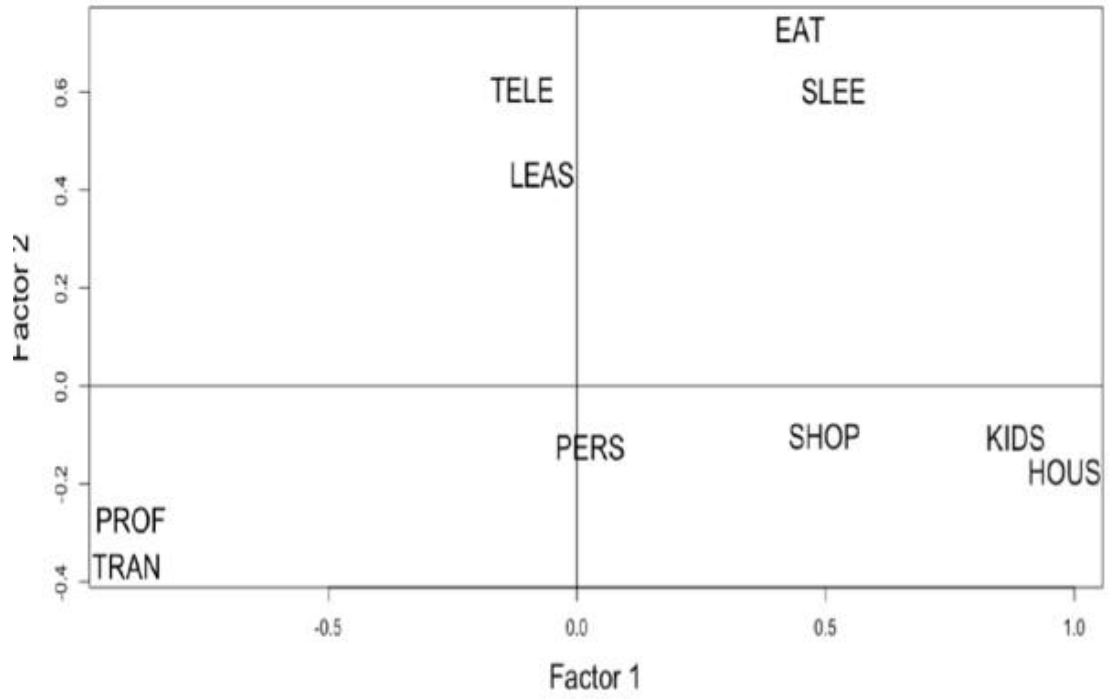

Figure 12: Representation of initial variables in "factor 1 factor 2" space

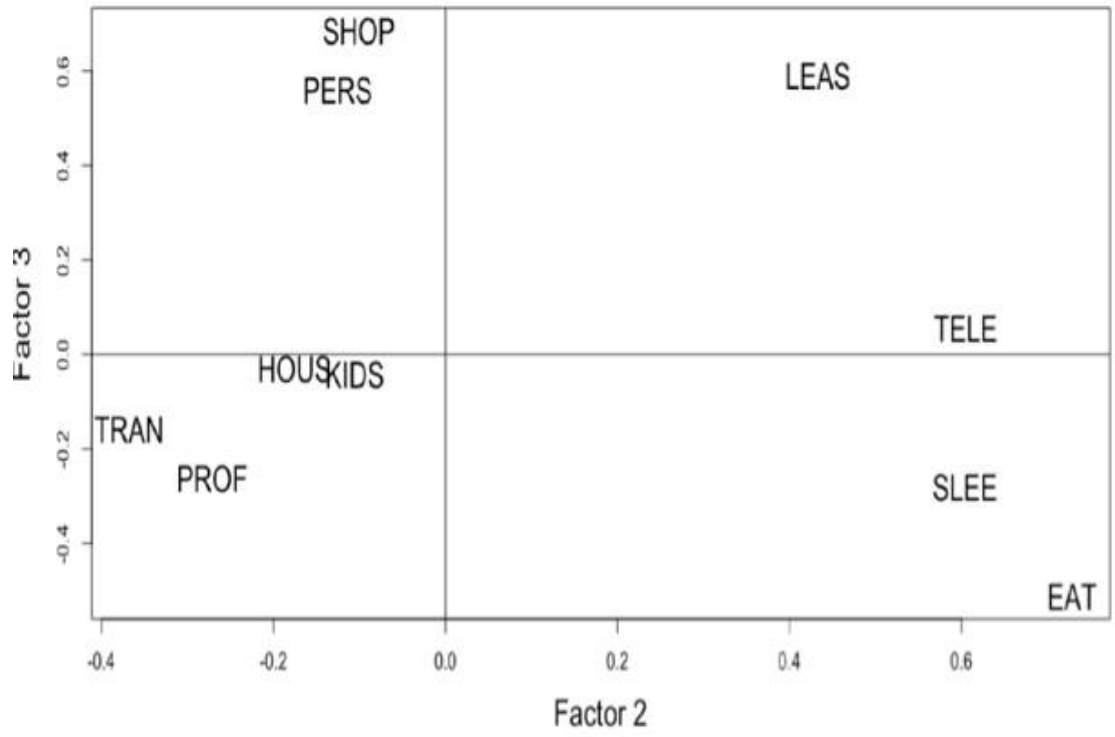

Figure 13: Representation of initial variables in "factor 2 factor 3" space

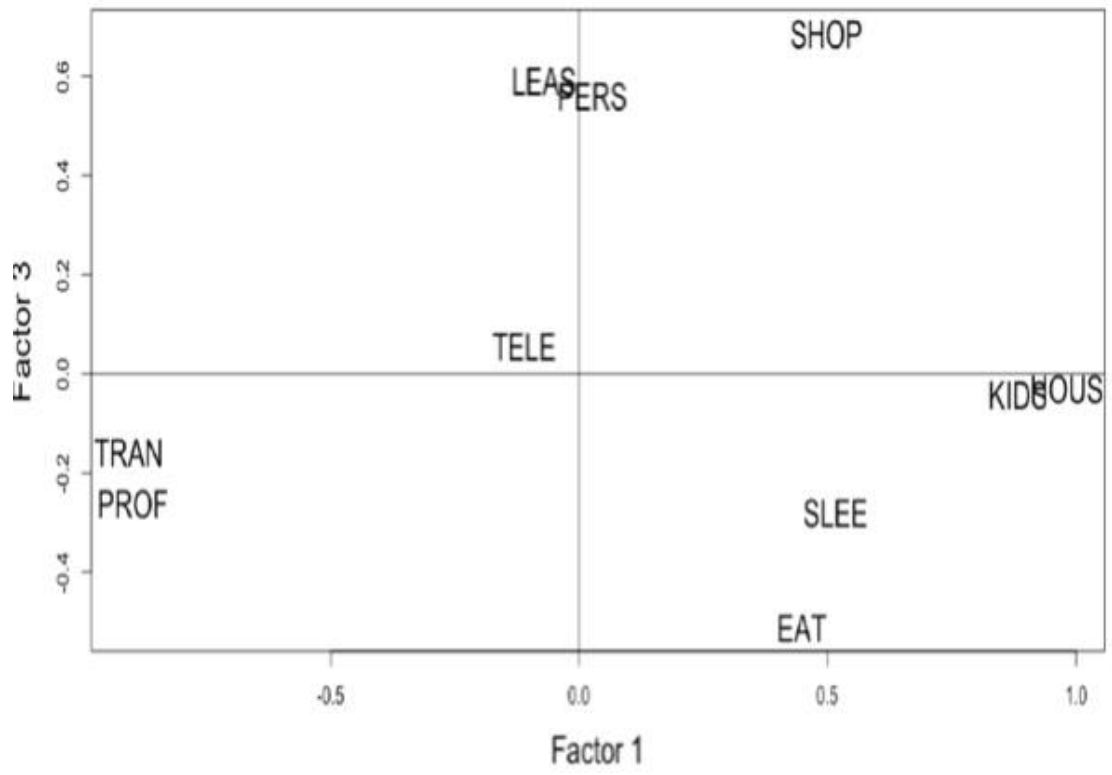

Figure 14: Representation of initial variables in "factor 1 factor 3" space

From the table 3 and figures 11-13 it appears that variables "TRAN", "PROF"and "HOUS", "KIDS" are strongly contrasted in the first factor. Thus, we may conclude that the least is bipolar (professional occupation is opposed to occupation connected with family), so we can call it "**Job VS family**". The second factor has the strongest connection with variables "EAT", "SLEE", "TELE", "LEAS". Consequently, we can call it "**physical rest/recreation**". The third factor can be, on the contrary, called "**emotional rest/self-nurturing**" as far as the biggest factor loadings in it belong to "SHOP", "PERS", "EAT".

4 Multivariate Regression

Regression analysis has long been used in many fields of research as a statistical tool for investigation of relationships between variables and building predictive models. Regression analysis usually includes finding causal effects of one variable upon another, estimating the quantitative effect of the predictors on the dependent variable, assessing statistical significance of these relationships, etc.

We apply multivariate regression analysis to our data set in order to attempt prediction of one dependent variable TELE (amount of hours spent watching television) based on other variables in 'Timebudget'. It must be noted that we had a rather unfortunate initial choice of the data set as far as regression analysis is concerned, as its variables are complementary by structure and all entries add up to 2400 hours total in each row (100 days spent by each person on listed activities monitored 24 hours a day). In this situation, a multivariate regression including all variables would result in a perfect fit, which makes the whole research method application rather pointless. We can prove this by attempting to build a full model with the *lm( )* command and checking the VIF (variance inflation factor) with the *VIF( )* command from the 'fmsb' package:

```
attach(Timebudget2)
Model1 <- lm(TELE ~ PROF+TRAN+HOUS+KIDS+SHOP+PERS+EAT+SLEE+LEAS,
data=Timebudget)
library('fmsb')
VIF(Model1)
summary(Model1)
```

The variance inflation factor in an index that measures how severe the multicollinearity problem in an OLS regression is, or, in other words, how much of the variance in the estimated coefficients of the regression can be attributed to collinearity in variables.

The *VIF( )* command results in [1] Inf output, indicating that the index is close to infinity whereas the normal indicator should not exceed 2(or 5 in other sources). Further diagnostics through the *summary()* of the model give us more evidence: the R-squared statistic is 1, all coefficients statistically significant with an alpha-value threshold at 0.001. In other words, attempting regression analysis on an essentially perfect fit appears to be pointless - any 9 variables of the data set will always result in a perfect forecast for the remaining one.

Yet, we do not let ourselves be discouraged by these results, and (keeping in mind that the

major goal of studying is ‘to fail forward through success‘) try using multivariate regression techniques on the given data set if not for the sake of a ‘good fit‘, than for the sake of learning. For this purpose, we create a shortened version of the data set to eliminate the complementarity problem, and build a ‘partial‘ regression with the use of a limited number of variables. But what variables fit best to predict the amount of hours spent on television? We take a look at the interdependencies between the variables with the help of the ‘*corrplot*‘ R package.

```
library('corrplot')
c <- cor(Timebudget)
corrplot(c, method = "circle")
```

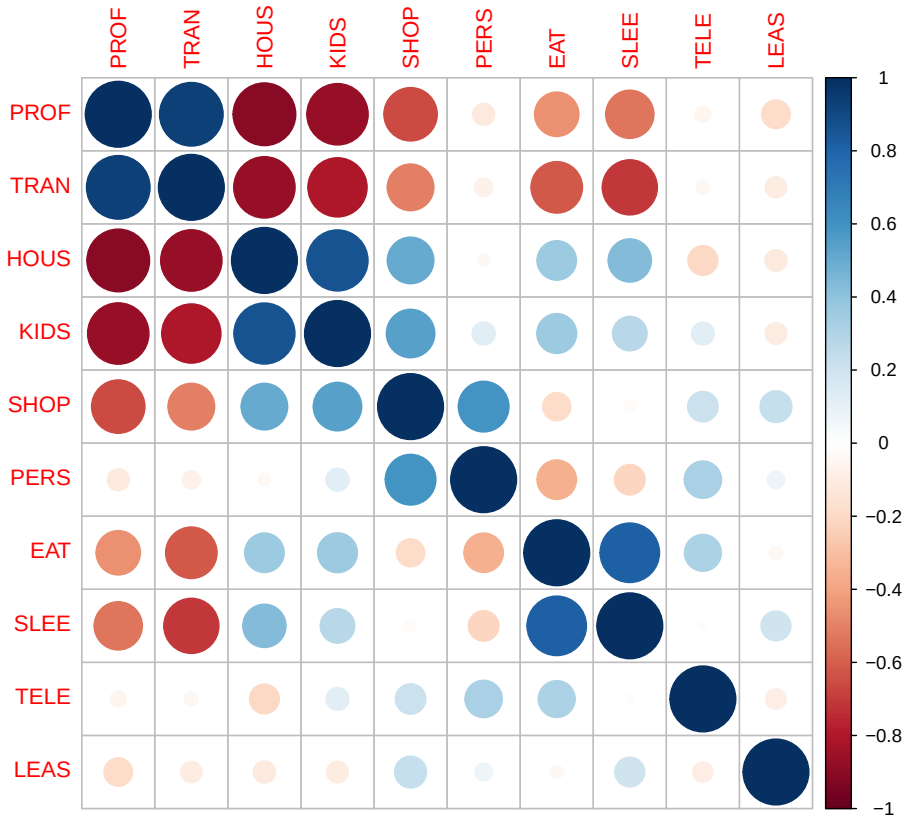

Figure 15: test

The graph shows the correlation coefficients of different variable pairs, with negative correlation pictured in dark red and positive correlation pictured in dark blue. None of the

variables seem to depict any sufficiently large correlation with our dependent variable TELE. However, it is clear that professional activities (PROF) are strongly positively correlated with hours spent on transportation (TRANS), and strongly negatively with time spent with children (KIDS) and house work (HOUS), which is something one could expect. We can take a more precise look at the corresponding relationships by using the '*pairs( )*'function, which constructs scatterplots of pairs of variables, and see that time spent on work and transportation have almost a perfect linear pattern with a positive slope. The same accounts for time spent on house work and time spent with children. The correlation of KIDS and SHOP seems less strong, but is still clearly visible from the plot.

```
pairs(~PROF+TRAN+HOUS+KIDS+SHOP)
```

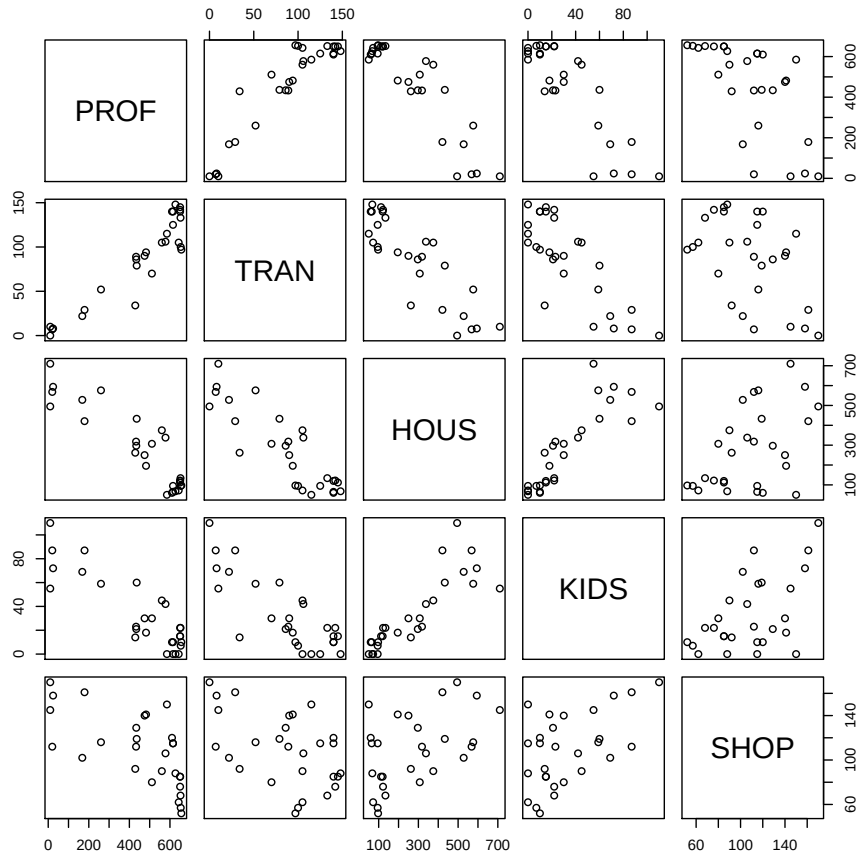

Figure 16: test

The findings suggest that the five variables may cause a multicollinearity problem if added to the regression model, so we exclude all of them in our first trial, and come up with the

following multivariate regression:

```
Model2 <- lm(TELE ~ PERS+EAT+SLEE+LEAS, data=Timebudget)
summary(Model2)
```

Call:

```
> lm(formula = TELE ~ PERS + EAT + SLEE + LEAS, data = Timebudget)
```

Coefficients:

	Estimate	Std. Error	t-Value	Pr(>—t—)
(Intercept)	627.80326	225.19594	2.788	0.010459*
PERS	1.96681	0.50866	3.867	0.000783***
EAT	1.98163	0.41296	4.799	7.69e-05***
SLEE	-1.23700	0.34980	-3.536	0.001764**
LEAS	0.06652	0.09316	0.714	0.482393

Table 5: test

Signif. codes: 0 ‘***‘0.001 ‘**‘0.01 ‘*‘0.05 ‘.‘0.1 ‘‘1

Residual standard error: 28.12 on 23 degrees of freedom

Multiple R-squared: 0.5664, Adjusted R-squared: 0.4909

F-statistic: 7.51 on 4 and 23 DF, p-value: 0.0005045

The first thing to notice from the R ‘*summary( )*‘ output is that the Multiple R-squared and Adjusted R-squared are relatively low - in other words, the model can account for only roughly half of the variance in the dependent variable. Secondly, the LEAS variable is not statistically important for the regression, the probability of ~ 0.48 clearly exceeding the 0.05-treshold, therefore we can discard it in future models. The well-known Gauss-Markov theorem states that the Ordinary-Least-Squares regression (OLS) leads to a Best Linear Unbiased Estimator (BLUE) results for the coefficients if, and only if the errors have zero expectation, are uncorrelated, and have equal variance. In order to prove that the resulting coefficient estimations are BLUE, we will check the five crucial regression assumptions:

1. Linearity

2. Normal distribution of residuals

3. Homoscedasticity of residuals

4. No multicollinearity in predictors

5. No autocorrelation in errors

We start with building diagnostic plots though a simple '*plot( )*'command:

```
layout(matrix(c(1,2,3,4),2,2))
plot(Model2)
```

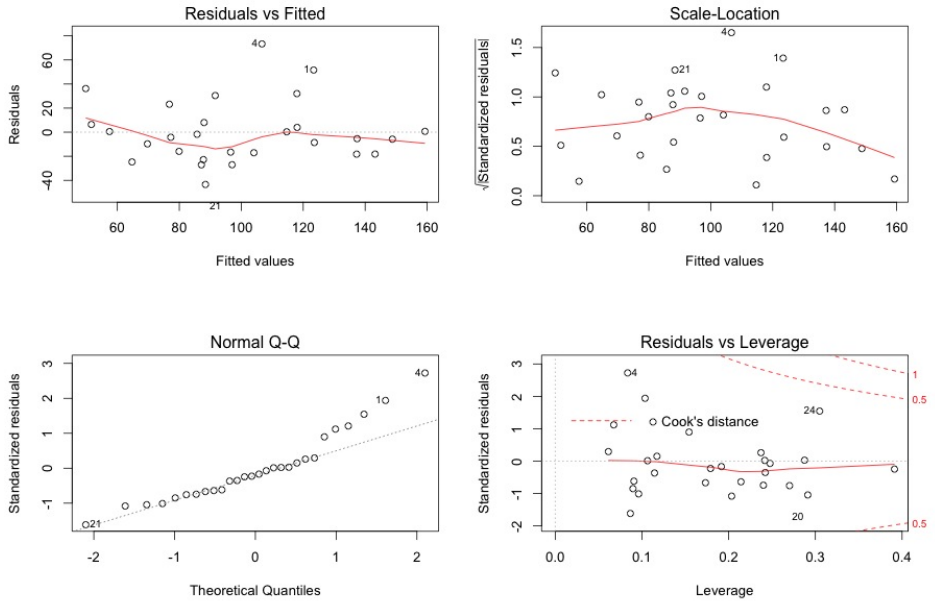

Figure 17: test

The first graph shows residuals vs. fitted values of the regression, the relatively straight line proves that the linearity assumption of the regression analysis is met. The panel on the lower left-hand side depicts a QQ-plot (standardized residuals vs. theoretical quantiles of a normal distribution) that serves for checking the 'Normal distribution' condition for residuals. This assumption seems to be violated, as the line goes noticeably up in the upper right-hand corner of the graph; moreover, three observations (1, 4, 21) are marked as outliers. The Scale-location plot allows us to check if residuals are spread equally along the ranges of predictors - again we notice three outliers, but the spread is otherwise equal across the entire

range (we allow for some curvature due to scarcity of observations). The final plot 'Residuals vs. Leverage' serves to identify influential outliers, that are usually positioned beyond the dashed lines along the edges of the plot representing high Cook's distance scores. We find that none of the outlier cases are influential to the regression results, which means that they are not important for our analysis.

Cooks distance of influential observations can be graphed with the the same command: '*plot(Model2, which=4)*'. Three outliers are marked in numbers:

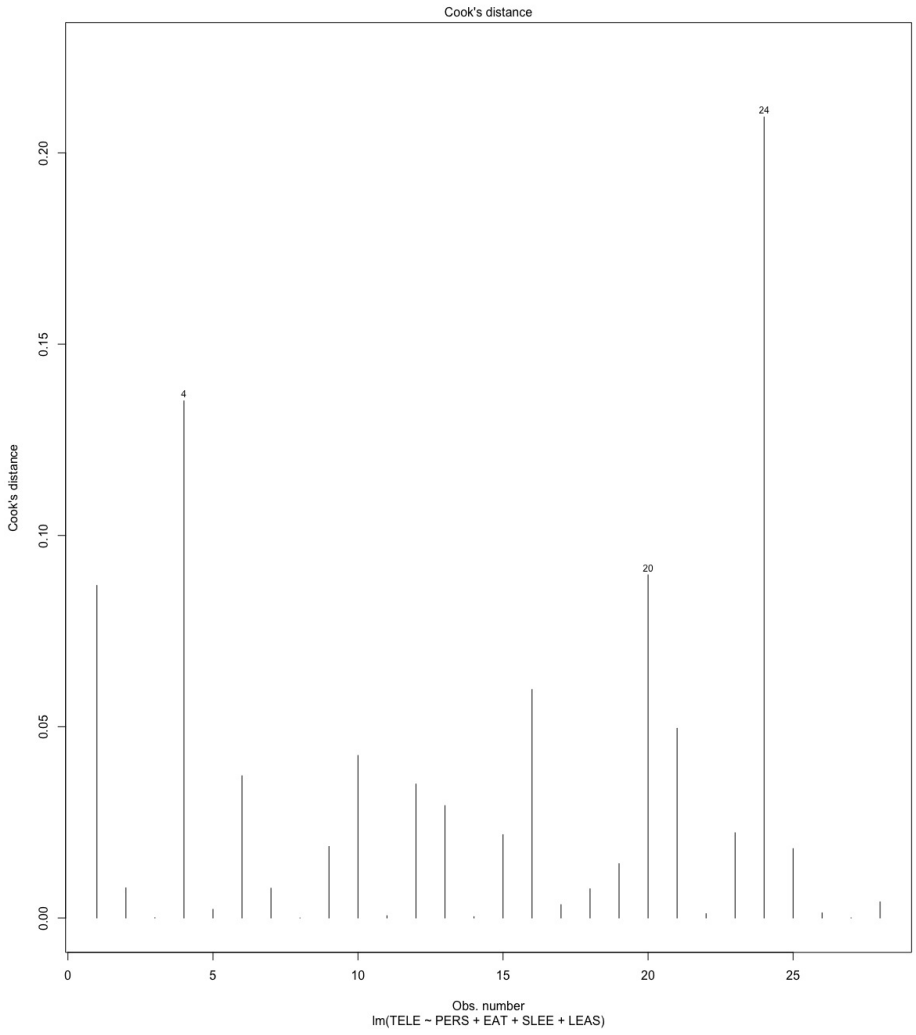

Figure 18: test

Apart from the basic regression diagnostics plot, R presents multiple options for assessing residual outliers and checking each individual condition separately. The '*car*' package function called '*outlierTest( )*' performs a Bonferroni outlier test and reports p-values for the most extreme observations that show how likely it is to get these values given a specific prob-

ability model. The Bonferroni adjustment multiplies the usually applied two-sided p-value by the number of observations. The results of the test show that no significant outliers are present:

```
library('car')
outlierTest(Model2)

# No Studentized residuals with Bonferonni p >0.05
# Largest —rstudent—:
# rstudent unadjusted p-value Bonferonni p
# 4 3.236291 0.0037931 0.10621
```

Further outlier detection (as well as proving normality assumption) can be done via '*car::qqPlot( )*'. The '*car::leveragePlots( )*' can be useful while showing the slope of the regression line build for each of the predictors individually, thus visualizing leverage of potentially important outliers:

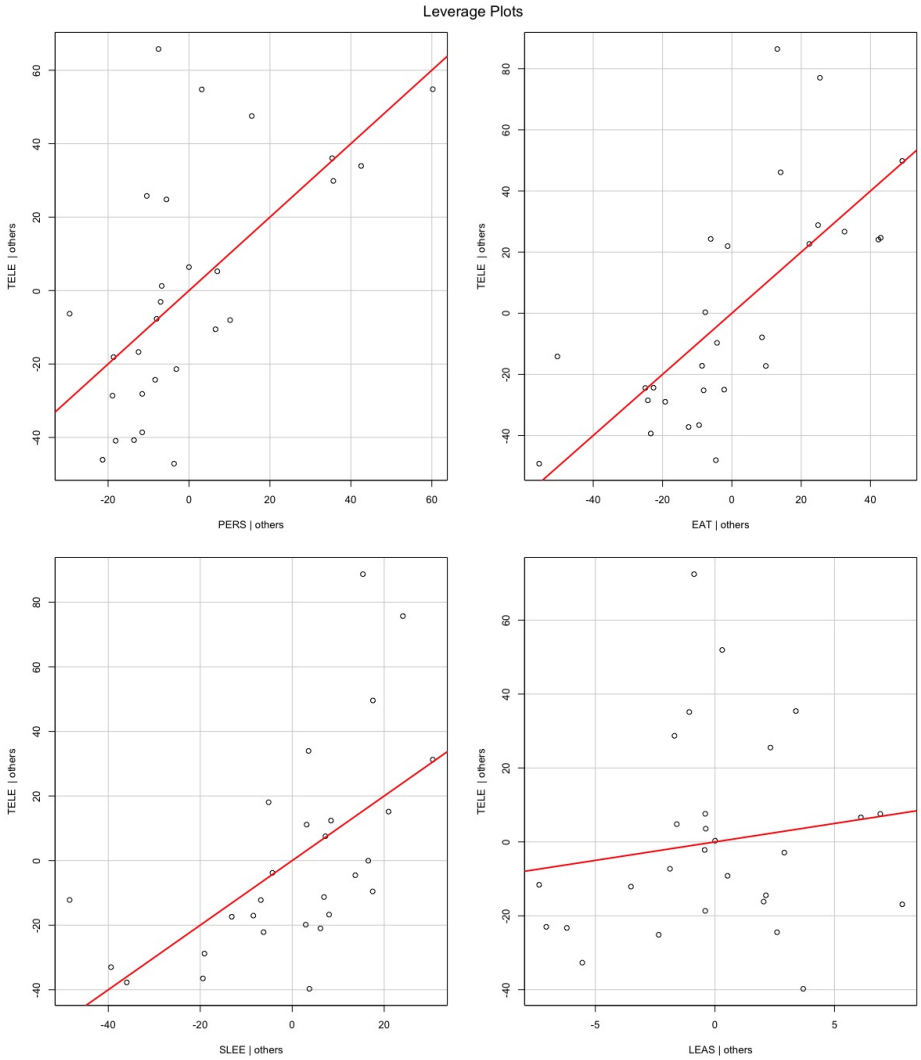

Figure 19: test

A further command is '*influencePlot( )*' that presents both outlyingness, leverage, as well as influence of each point. The residuals are shown on the vertical axis, leverage is marked on the horizontal axis, the point size is determined by Cook's Distance. Although being a somewhat more complex construction than previous plots, it helps convey the previously obtained information in a more intuitive manner:

```
influencePlot(Model2, id.method="identify", main="Influence Plot", sub="Circle s
```

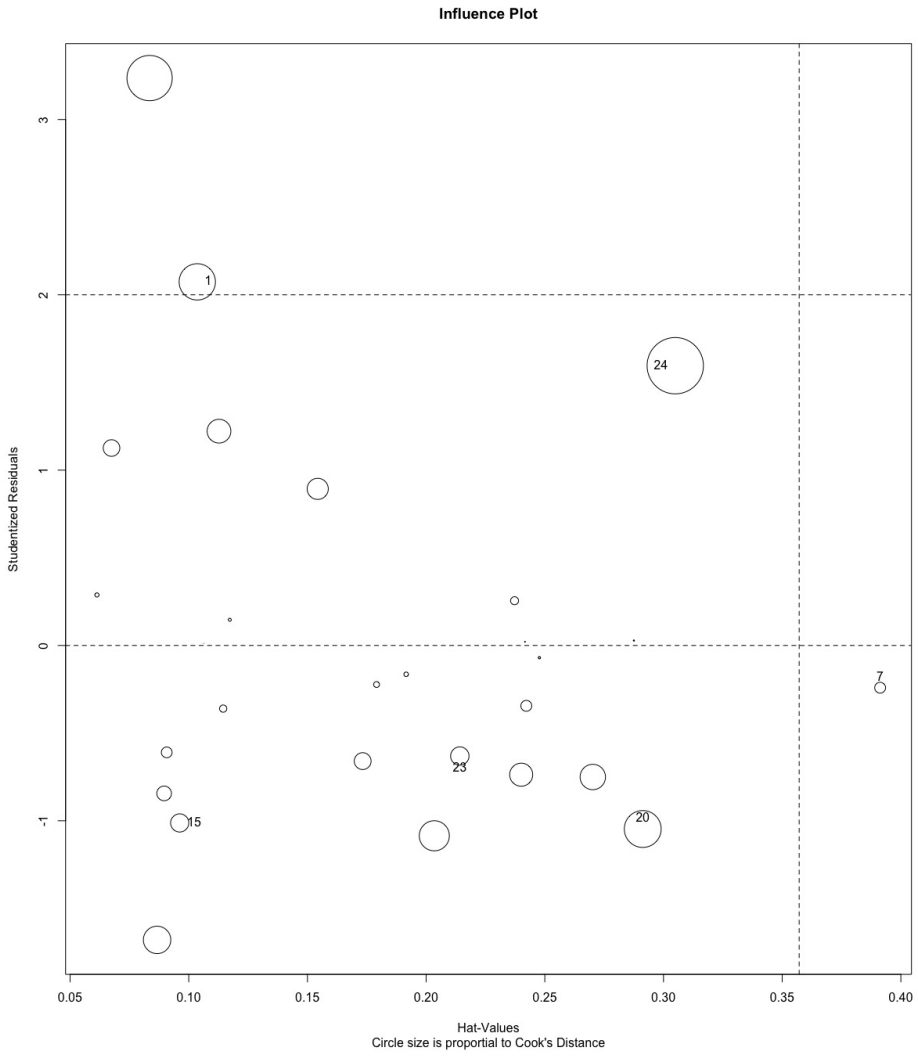

Figure 20: test

The 'MASS'package in R can be used for proving the normality of residuals through a histogram. We find that the residuals are normally distributed, but the distribution is somewhat skewed.

```
library(MASS)
sresid <- studres(Model2)
hist(sresid, freq=FALSE, main="Distribution of Studentized Residuals", breaks=10
xfit <- seq(min(sresid),max(sresid),length=40)
yfit <- dnorm(xfit)
lines(xfit, yfit)
```

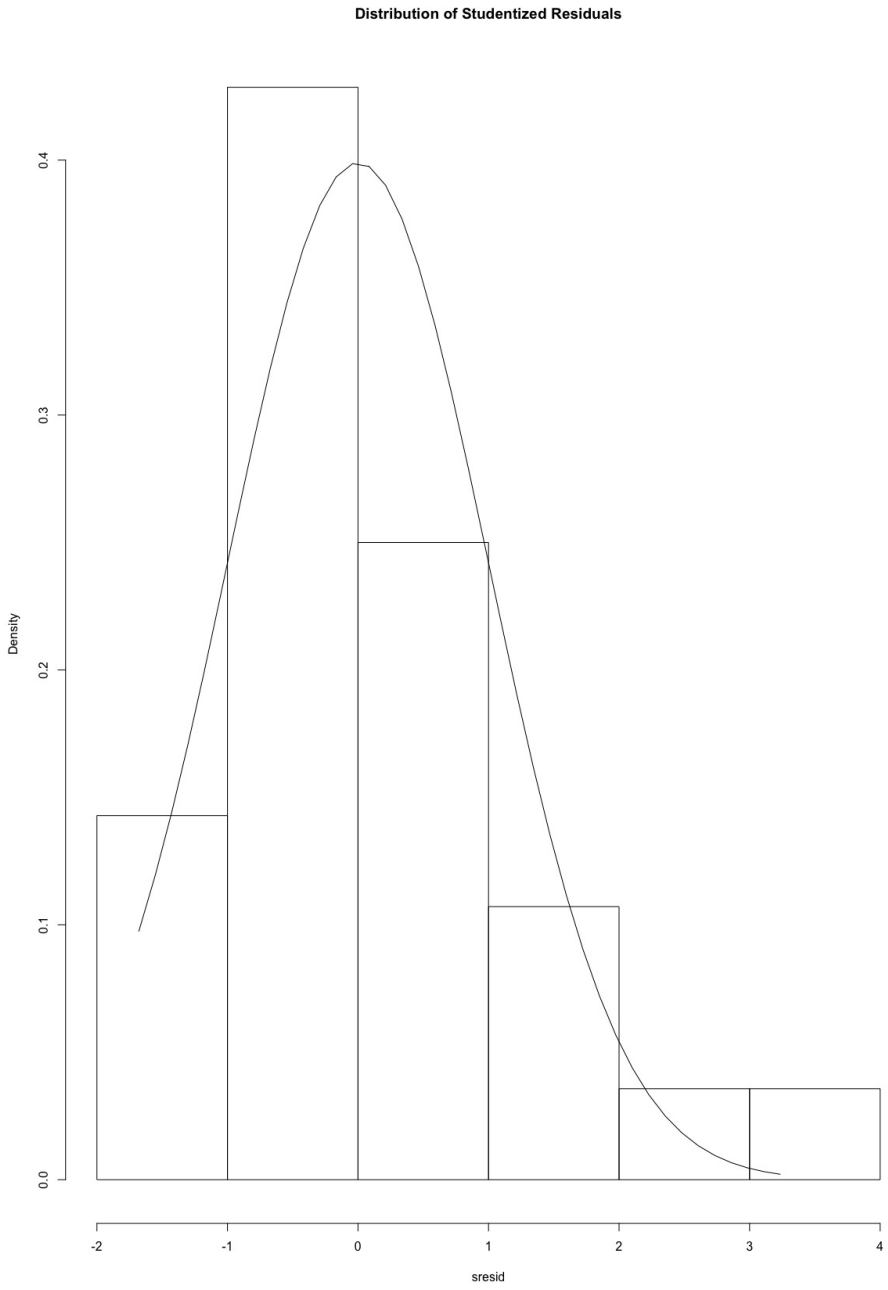

Figure 21: test

We then proceed with testing for non-constant error variance (Homoscedasticity) using the Breush-Pagan Test, which is carried out with the '*ncvTest( )*' command. In a standard linear regression, the variance of the residuals must be constant (independently distributed) over the values of the response variable (fitted values). The null-hypothesis of the Breush-Pagan test assumes homoscedasticity. We find that the errors are, with a large p-value (p = 0.998919), homoscedastic. After this, we evaluate collinearity with the aforementioned VIF index ('*VIF( )*' command), and find that the variance inflation factor of our model is 2.306042

with the '*sqrt(VIF(Model2)) >2*'command giving a [1] FALSE response. This indicates that no multicollinearity is present (it is a widely accepted heuristic that the squared root of VIF should not exceed 2). The conditions are met.

Another major assumption in a linear regression is that the dependence of the predicted variable on each of the predictors is linear. We evaluate non-linearity with two commands: '*crPlots( )*'and '*ceresPlots( )*'. The latter command is called the 'CERES plot'and was developed by Dennis Cook to depict a curve in the relationship of the dependent variable to a given predictor, even if nonlinear relationships among the predictors are present. We can compare these plots to the previously made leverage plots and find that some non-linearity is present (see Fig. N) . However, it is hard to draw a definitive conclusion given the small sample size of 28 observations that we base our research on.

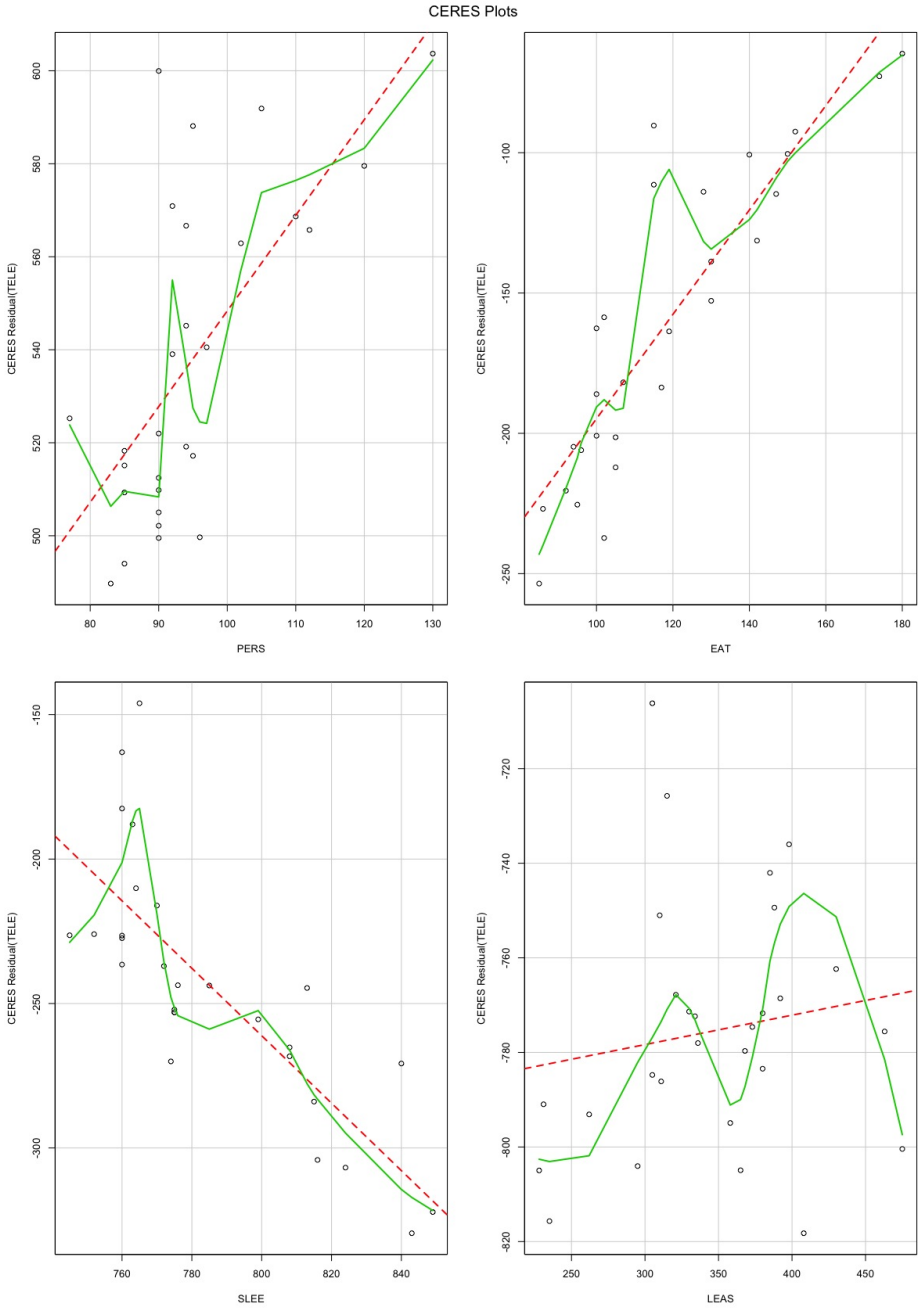

Figure 22: test

Finally, the test for autocorrelation is carried out with the function '*durbinWatsonTest( )*'. On the basis of the results we conclude that there is no autocorrelation in errors in our model.

lag Autocorrelation D-W Statistic p-value

1 0.09093127 1.669289 0.246

Alternative hypothesis: rho != 0

Another extremely useful package in R is the '*gvlma*'package (Global Validation of Linear Models Assumptions) that allows us to automatically perform test on the models assumptions via '*summary( )*'command. The skewness and the kurtosis measures relate to the shape of the distribution of the residuals. As expected, the distribution is not a non-skewed normal distribution, and does not pass the test. We have seen this in our previous graphs 'Distribution of studentized residuals'and the 'QQ plot'.

```
library('gvlma')
gvmodel2 <- gvlma(Model2)
summary(gvmodel2)
```

#	Value	p-Value	Decision
# Global Stat	8.29836	0.08124	Assumptions acceptable!
# Skewness	4.94670	0.02614	Assumptions NOT satisfied!
# Kurtois	0.87660	0.34913	Assumptions acceptable!
# Link Function	0.04419	0.83350	Assumptions acceptable!
# Heteroscedasticity	2.43087	0.11897	Assumptions acceptable!

In order to achieve better predictive power, we add each of the previously discarded variables that involved multicollinearity separately to our first model to create five other regression models:

```
Model3Prof <- lm(TELE ~ PROF+PERS+EAT+SLEE+LEAS, data=Timebudget)
Model3Tran <- lm(TELE ~ TRAN+PERS+EAT+SLEE+LEAS, data=Timebudget)
Model3Hous <- lm(TELE ~ HOUS+PERS+EAT+SLEE+LEAS, data=Timebudget)
Model3Kids <- lm(TELE ~ KIDS+PERS+EAT+SLEE+LEAS, data=Timebudget)
Model3Shop <- lm(TELE ~ SHOP+PERS+EAT+SLEE+LEAS, data=Timebudget)
```

The regression summary statistics prove that LEAS as a variable is statistically insignificant in all model settings, therefore we discard of it completely. Model3Hous (involving time spent on home work as one of the predictors) produces the best Multiple R-squared (0.6389) and the best Adjusted R-squared (0.5568). The model with four predictors (HOUS, PERS, EAT, SLEE) produced even better results with adjusted R-squared of 0.5751. To improve further our model's predictive power, and just for the sake of curiosity, we created

manually two dummy variables based on the additional demographic information provided by the dataset‘s description about the 28 people that took part in the research. Our dummies include GEND (Gender: female = 1, male =0) and GEO (Geographic location: person from a western country or the US = 1, person from eastern country or Yugoslavia = 0). The final model looked the following way:

summary(ModelDummy)

Call:

lm(**formula** = TELE ~ PERS + EAT + SLEE + HOUS + GEND + GEO, **data** = TimebudgetD)

Residuals:

Min	1Q	Median	3Q	Max
-37.064	-10.158	-4.036	7.090	39.654

Table 6: test

Coefficients:

	Estimate	**Std. Error**	**t-value**	**Pr(¿—t—)**
(Intercept)	289.29253	164.74688	1.756	0.09367
PERS	1.32578	0.64498	2.056	0.05247
EAT	0.28662	0.52155	0.550	0.58843
SLEE	-0.48595	0.24854	-1.955	0.06400
HOUS	0.14117	0.04994	2.827	0.01010*
GEND	-65.53088	17.75841	-3.690	0.00136**
GEO	61.34802	18.60622	3.297	0.00343**

Table 7: test

– – –

Signif. codes: 0 ‘***‘0.001 ‘**‘0.01 ‘*‘0.05 ‘.‘0.1 ‘‘1

Residual standard error: 19.11 on 21 degrees of freedom

Multiple R-squared: 0.8171, Adjusted R-squared: 0.7649

F-statistic: 15.64 on 6 and 21 DF, p-value: 8.924e-07

The model yielded very good predictive results with a Multiple R-squared of 0.8171, Adjusted R-quared of 0.7649, and a F-statistic p-value of 8.924e–07. It is remarkable, that two newly introduced dummy variables had the best p–values for statistical significance at a 0.01 value threshold. The variable EAT turned out to be insignificant for the regression model, PERS and SLEE - significant only at alpha = 10%. Reducing the model by a further variable EAT produced the highest adjusted R-squared – 77,2%.

We check our model assumptions with the same methods as described before for Model2. The resulting graphs can be found in the Appendix. The four diagnostics plots (see Fig N (plot1)). The plots show possible condition violations in model linearity, and three outliers (numbers 4, 20, 25). The normality assumption violation is also open to question - the distribution of residuals seems to suffer from the same skewness as in the previous models. The Bonferonni outlier test (‘*car:: outlierTest( )*‘) shows no studentized residuals with Bonferonni p <0.05. The QQplot (plot 2) supports the finding by proving that all points lie within the dotted line serving as a threshold for non–significant outliers. The normality assumption also holds here, as the studentized residuals‘ positions correspond to the normal distribution quantlets. Plot 4 (see App), however, shows a more contradictory picture regarding the non–violation of the normality assumption: the skewness to the left is clearly noticeable here. It is therefore hard to make any steadfast conclusions regarding the residuals‘ distribution given that we still have a small sample size.

The ‘*ncvTest( )*‘ shows us no heteroscedasticity in residuals. There is, however, a problem with multicollinearity:

```
sqrt(VIF(ModelDummy2)) > 2
```

```
[1] TRUE
```

Plot 5 depicts correlation coefficients of pairs of variables used in our model. It seems that House work activities (HOUS) have a strong positive correlation with Gender (GEND), which contributes to the VIF index and violates the non-multicollinearity assumption.

The model linearity assumption is checked again the the CERES plots (plot6). The results, again due to sample size, are hard to interpret in a non-contradictive way, but we take the risk deciding that no strong non-linear patterns are present in the data. Finally, the ‘gvlma::summary‘gives us the following output:

ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS

USING THE GLOBAL TEST ON 4 DEGREES–OF–FREEDOM:

Level of Significance = 0.05

Call:

```
gvlma(x = ModelDummy2)
```

#	Value	p-Value	Decision
# Global Stat	4.25505	0.37259	Assumptions acceptable!
# Skewness	0.447960	0.50331	Assumptions acceptable!
# Kurtois	0.05135	0.83074	Assumptions acceptable!
# Link Function	3.52164	0.06057	Assumptions acceptable!
# Heteroscedasticity	0.234100	0.62850	Assumptions acceptable!

Having checked all necessary regression model assumptions and having come to a satisfactory result, we conclude that the multivariate regression model based on five major predictors (time spent on personal activities, sleep, house work, gender and geographic location) contributes the best possible results for predicting our dependent variable TELE - time spent watching television. The results seem plausible and intuitively correct, so we pick the dummy-containing model as the best predictive model generated in our research and move on to Factor analysis.

References

Breusch, T. S. and P. Schmidt (1988): "Alternative Forms of the Wald test: How Long is a Piece of String," *Communications in Statistics, Theory and Methods*, 17, 2789–2795.

Gallant, A. R. (1987): *Nonlinear Statistical Models*, New York: John Wiley & Sons.

A Figures

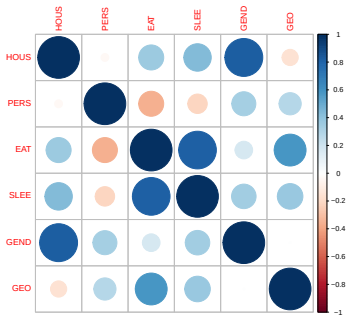

Figure 23: test

B Tables